Thesis/
Reports
Bevins,
C. D.

RETENTION COPY

Edition FEB 1987
Item No. 73
File Coop Agreement

FY86 FINAL REPORT FOR COOPERATIVE AGREEMENT No 22-C-6-INT-041
with SYSTEMS FOR ENVIRONMENTAL MANAGEMENT
FS Contact: R. ROTHERMEL
Co-op Contact: C. Bevins

# P R O G R A M  B L D M D L

## A Prototype System to Select Fuel Models Based Upon Fire Observation Data

Progress Report No. 1
Cooperative Aid Agreement No. 22-C-6-INT-041
November 1986

Submitted by
Collin D. Bevins, Principal Investigator
Systems for Environmental Management

Submitted to
Richard C. Rothermel, Project Manager
Intermountain Fire Sciences Laboratory

*SYSTEMS FOR ENVIRONMENTAL MANAGEMENT*

P.O. BOX 8868
MISSOULA, MONTANA 59807
PHONE: (406) 549-7478

## *SYSTEMS FOR ENVIRONMENTAL MANAGEMENT*

A Natural Resources Research Group

Systems for Environmental Management (SEM) is a non-profit research and educational corporation based in Missoula, Montana, a regional center for natural resources agencies. Founded in 1977, SEM works cooperatively and under contract with the U.S. Forest Service, National Park Service, Bureau of Land Management, and Bureau of Indian Affairs, as well as state and academic institutions, private individuals and organizations.

SEM's diverse professional staff offers a wide range of research capabilities within the natural resources management field. The full-time staff includes specialists in fire behavior, fire history, recreation management, geography, plant ecology, forestry, meteorology, data analysis, and computer science. SEM also maintains a pool of professional affiliates that can be utilized for projects requiring additional expertise. Areas of recent research emphasis include wilderness recreation management, fire history and ecology, fire planning, and development of computerized resource management tools.

Property of
National FS Library
USDA Forest Service
240 W Prospect Rd
Fort Collins CO 80526

# P R O G R A M  B L D M D L

## A Prototype System to Select Fuel Models Based Upon Fire Observation Data

Progress Report No. 1
Cooperative Aid Agreement No. 22-C-6-INT-041
November 1986

Submitted by
Collin D. Bevins, Principal Investigator
Systems for Environmental Management

Submitted to
Richard C. Rothermel, Project Manager
Intermountain Fire Sciences Laboratory

## PREFACE

Progress Report Number 1 is submitted in partial fulfillment of Cooperative Aid Agreement No. 22-C-6-INT-041, Section II.C, entered into between Systems for Environmental Management, a nonprofit research corporation located in Missoula, Montana, and the Intermountain Forest Experiment Station, Ogden, Utah.

## INTRODUCTION

In 1972, R.C. Rothermel proposed a mathematical model to predict fire spread rate and flame length given a set of fuel and environmental descriptors. The model has since been field tested and incorporated into many fire behavior and danger rating systems. Notable among these are the National Fire Danger Rating System and the BEHAVE System.

Both the NFDRS and BEHAVE utilize the concept of fuel models to describe the physical and biological properties of the fuel complex. To predict fire behavior, one need only specify a fuel model, terrain slope, and the wind and moisture conditions of the site.

The NFDRS uses a limited set fuel models to rate fire danger, and BEHAVE employs the thirteen standard "NFFL" fuel models. BEHAVE also introduced the concept of the custom fuel model, where system users supply the fuel model parameters for a specific site. The fuel parameter values are taken from field inventories under the assumption that real-world fuel inputs produce reasonable fire model predictions.

The above process of building fuel models is often a hit-and-miss chore (Omi, 1986). The fuel modeler could explore many input parameters with an infinite set of values. Oftentimes the modeler is attempting to select a set of fuel inputs that yield fire behavior predictions in agreement with field observations.

The objective of the current study is to examine the feasibility of using a computer to construct fuel models which best predict observed fire behavior. The study has so far focused on the development of algorithms which accept a set of fire behavior, moisture, wind, and slope observations as inputs, and systematically examines alternative sets of fuel parameters (e.g., fuel models), selecting the one(s) which will best predict the observed fire behavior.

The ultimate study objective is to develop a user-friendly system, in the spirit of BEHAVE, which builds custom fuel models from fire observations. This would be a valuable tool for fire researchers and managers who could then use a limited number of fire observations (such as from prescribed fires) from a limited set of environmental conditions to construct a well-behaved fuel model. The fuel model could then be used with BEHAVE to make reasonable fire behavior projections under environmental conditions outside those of the field observations.

## THEORY AND ALGORITHMS

### Mathematical Model Parameters

The Rothermel (1972) fire model requires the following input parameters;

1 fuel particle heat of combustion (BTU/lb),
2 fuel particle density (lbs/cuft),
3 fuel particle mineral content (fraction),
4 fuel particle silica-free mineral content (fraction),

5 fuel load (lb/sqft) by category and size class,
6 fuel particle surface area-to-volume ratio (sqft/cuft) by category and size class,
7 fuel bed depth,
8 fuel moisture of extinction (fraction ovendry weight),

9 terrain slope,
10 midflame wind velocity (ft/min), and.
11 fuel particle moisture content (fraction ovendry weight) by category and size class.

The first eight parameters describe the fuel condition and constitute a fuel model. The first four parameters are usually treated as constants across all fuel types, categories, and size classes. This leaves fuel load, surface area-to-volume ratio, bed depth, and extinction moisture as the parameters characterizing fuel models.

Fuel load, surface area-to-volume ratio, and fuel moisture are reduced by the mathematical model to a single 'characteristic' value using surface area weighting techniques. The mathematical fire model internally uses only the following intermediate fuel parameters;

1 characteristic fuel surface area-to-volume ratio (SIGMA),
2 fuel bed packing ratio (dimensionless) (BETA),
3 characteristic net fuel load (WN), and
4 extinction moisture (MEXT).

An algorithm which tests the ability of a wide range of fuel parameter combinations to predict a set of fire behavior observations therefore needs only to incrementally vary the above four parameters over some reasonable range of values.

## Selection of Alternative Fuel Models

A method is needed to select the 'best' fuel model parameter set from all examined alternatives. Each fuel parameter set is used to predict each fire behavior observation using the observation's slope, wind, and fine fuel moisture values. A least squares regression is then performed of the predicted vs. observed fire behavior:

$$\text{Observed} = a + b * \text{Predicted} + \text{SEE}$$

where:

Observed = the observed fire behavior (spread rate or flame length)
a = regression intercept,
b = regression slope,
Predicted = the predicted fire behavior (spread rate or flame length), and
SEE = the standard error of the estimate.

The slope 'b' is an adjustment factor to the predicted fire behavior. The characteristic net load (WN) can be factored out of the mathematical fire model. The regression slope 'b' can therefore be used to derive a WN which yields a slope of 1.

This has two advantages. First, WN is derived and can be dropped from the iterative search algorithm. Second, the slope is forced to unity and another regression characteristic, such as minimum SEE, maximum correlation coefficient, or regression intercept nearest zero, can be selected as a test statistic.

## Fuel Parameter Search Algorithm

The following pseudocode describes the algorithm developed to test alternative fuel model parameter sets;

```
.1  For each SIGMA value within some minimum, maximum, and increment,
2   For each BETA value within some minimum, maximum, and increment,
3   For each MEXT value within some minimum, maximum, and increment,
4         Set WN = 1.
5         For each fire behavior observation,
6               Use SIGMA, BETA, MEXT, WN, and environmental observations to
                predict the fire behavior observation.
7         Derive least squares statistics of observed vs predicted behavior.
8         Set WN = b * WN.  (For flame length, use WN = b**(1.087) * WN)
9         For each fire behavior observation,
10              Use SIGMA, BETA, MEXT, WN, and environmental observations to
                predict the fire behavior observation.
11        Derive least squares statistics of observed vs predicted behavior.
12        Derive test statistic for this SIGMA-BETA-MEXT-LOAD fuel model.
13        Insert the fuel model and its regression statistics into results table
          based upon test statistic value.
14  Next MEXT, BETA, or SIGMA.
15  Display ordered results table.
```

# BLDMDL TECHNICAL DOCUMENTATION

A computer program, BLDMDL, was written to codify and test the fuel parameter search algorithm. The program reads all its input data from a single disk file and prints processing information to the screen, and optionally, to a log file. The program was written in ANSI 1977 standard FORTRAN to facilitate porting between micro- and minicomputers.

## BLDMDL Inputs

BLDMDL reads its input from a file named 'BLDMDL.RUN'. The input file consists of a series of records containing an identifying keyword followed by a set of keyword-specific arguments. The file is read using free-field format, and records may appear in any random order. The following keywords are recognized:

BETA minimum value, maximum value, increment value

Optional. Specifies minimum, maximum, and incremental fuel bed packing ratio values to be tested. If omitted, defaults to 0.0010., 0.0500, and 0.0010., or 50 values. While any number of BETA records may appear in the file, only the last BETA record is used by BLDMDL.

DOC <documentation literal string>

Optional. Specifies a documentation literal string to be echoed to the console or log file (if invoked). Any number of DOC records may appear in the file.

END

Optional. Signals the end of the input file. Any records following the END record are ignored.

FIRE label, spread, flame, weight, slope, wind, fine fuel moisture

At least two FIRE records are required in the file. BLDMDL accepts up to 1000 FIRE records. FIRE records need not occur in a contiguous block. Each record specifies a single fire behavior observation set:

FIRE - the FIRE record keyword.
label - a 1 to 8 character label. Labels must be enclosed in single or double quotes if they contain embedded blanks.
spread - observed spread rate in feet per minute.
flame - observed flame length in feet.
weight - observation weighting factor. Must be an integer.
slope - terrain slope expressed as a fraction, not as a percent.
wind - midflame wind velocity in miles per hour.
fine fuel moisture - expressed as a fraction of ovendry weight.

FUEL <fuel description literal string>

Optional. BLDMDL does not currently use this information. It may be used in later versions to distribute SIGMA and WN values among fuel categories and size classes.

LOG <OFF | file name>

Optional. Specifies a processing log file. The log file is turned on at the command, and all remaining input, processing, and error information is echoed to the specified log file. The log file is turned off by specifying 'OFF' as the log file name. While BLDMDL accepts a file name up to 64-characters in length, the resident operating system may have other constraints.

MEXT minimum value, maximum value, increment value

Optional. Specifies minimum, maximum, and incremental extinction moisture values to be tested. If omitted, defaults to .10, .40, .05, or 7 values. While any number of MEXT records may appear in the file, only the last MEXT record is used by BLDMDL.

PREDICT <SPREAD | FLAME>

Optional. Specifies whether fuel model should use spread rate or flame length as the fire behavior to predict. Defaults to SPREAD.

SIGMA minimum value, maximum value, increment value

Optional. Specifies minimum, maximum, and incremental characteristic fuel particle surface area-to-volume ratio values to be tested. If omitted, defaults to 1000., 4000., and 100., or 31 values. While any number of SIGMA records may appear in the file, only the last SIGMA record is used by BLDMDL.

TEST <SEE | R | COMBINATION | INTERCEPT>

Optional. Specifies the regression statistic used to select the best fuel model parameter set. Legal arguments are:

SEE - select minimum standard error of the estimate
R - select maximum correlation coefficient
COMBINATION - selects minimum SEE + ABS(INTERCEPT)
INTERCEPT - selects intercept closest to zero

If omitted, defaults to SEE.

# BLDMDL Processing

BLDMDL Version 0.1 contains the following 16 program units and 3 include files;

ANALIZE.FOR (SUBROUTINE ANALIZE)

Subroutine called by FINDMODEL to derive fire behavior predictions for a specific fuel parameter set, perform least squares analysis, adjust WN to the regression slope, then derive a new set of behavior predictions and least squares statistics. It calls PREDICT, LEASTSQ, and TESTSTAT.

BEHAVE.FOR (SUBROUTINE BEHAVE)

Subroutine called by PREDICT to make a single fire behavior prediction. Calls no other program units.

BLDMDL.FOR (PROGRAM BUILDMODEL)

Main program unit. Calls, in order, integer functions READSPEC, FINDMODEL, MAKEMODEL, REPORTMODEL, and subroutine FINISH.

BLKDATA.INC

Unlabeled block data include file containing all labeled common initializations. Included into BLDMDL only.

COMMON1.INC

Labeled common block include file. Included by all program units except OPENFILE and WREGRESS.

COMMON2.INC

Labeled common block include file for BEHAVE parameters. Included by ANALIZE, BEHAVE, DISPLAY, FINDMODEL, INSERT, MAKEMODEL, PREDICT, and REPORTMODEL.

DISPLAY.FOR (SUBROUTINE DISPLAY)

Called by FINDMODEL to perform all processing output to console and log file. Calls no other program units.

FINDMODL.FOR (INTEGER FUNCTION FINDMODEL)

Called by BUILDMODEL to iteratively test alternative fuel model parameter sets. Calls BEHAVE, ANALIZE, DISPLAY and INSERT.

FINISH.FOR (SUBROUTINE FINISH)

Called by BUILDMODEL to close open files. Calls no other program units.

INSERT.FOR (SUBROUTINE INSERT)

Called by FINDMODEL to insert the current fuel model parameter set and its regression statistics into the results table. INSERT maintains the results table in sorted order according the specified test statistic. The best 100 fuel model sets are kept in the table. INSERT calls no other program units.

LEASTSQ.FOR (SUBROUTINE LEASTSQ)

Called by ANALIZE to perform least squares analysis on either the spread rate or the flame length predictions and observations. It calls subroutine WREGRESS to perform the least squares regression.

MAKEMODL.FOR (INTEGER FUNCTION MAKEMODEL)

Called by BUILDMODEL to partition selected fuel model parameters into fuel categories and size classes. Not operational in version 0.1. It calls no other program units.

OPENFILE.FOR (SUBROUTINE OPENFILE)

General purpose subroutine called by READSPEC to open and test files. It calls no other program units.

PREDICT.FOR (SUBROUTINE PREDICT)

Called by ANALIZE to make fire behavior predictions for all individual fire observation inputs. It calls BEHAVE.

READSPEC.FOR (INTEGER FUNCTION READSPEC)

Called by BUILDMODEL to read, test, and store user inputs. It calls OPENFILE and TESTRANGE.

REPTMODL.FOR (INTEGER FUNCTION REPORTMODEL)

Called by BUILDMODEL to report final results to the console and log file. It calls ANALIZE, DISPLAY, and LEASTSQ.

TESTRANG.FOR (INTEGER FUNCTION TESTRANGE)

Called by READSPEC to test and store user-supplied BETA, MEXT, and SIGMA fuel model parameter test values. It calls no other program units. Returns 0 if no errors discovered, otherwise it returns 1.

TESTSTAT.FOR (REAL FUNCTION TESTSTAT)

Called by ANALIZE to compute and return the test statistic for a specific least squares regression. The test statistic is determined by the TEST record argument at input time. It calls no other program units.

WREGRESS.FOR (SUBROUTINE WREGRESS)

A general purpose weighted regression subroutine called by ANALIZE and REPORTMODEL. It calls no other program units.

## BLDMDL Output

BLDMDL displays all inputs, processing information, and outputs to the display console, and optionally, to the specified log file. The output includes;

-- an echo of the input file, including error diagnostics,
-- a summary of SIGMA, BETA, and MEXT values to be examined,
-- a line of information about each fuel model parameter set as it is tested (sent to console only),
-- an ordered listing of the best 100 fuel parameter sets and their test values and regression statistics,
-- a table of the SIGMA, BETA, MEXT, LOAD, and DEPTH parameters for the best fuel model parameter set, and
-- a table each for spread rate and flame length predictions vs. observations using the best fuel model parameter set.

## BLDMDL Execution Requirements

The BLDMDL.EXE executable file for microcomputers running under MS-DOS requires about 111K of memory. Dynamic memory allocation is not used. The processing log file occupies 4-8K of disk space.

## TESTING

The BLDMDL fire behavior algorithms were tested with several control input data sets, and output was verified against BEHAVE predictions.

The BLDMDL fuel model selection algorithms were given preliminary testing in a two-stage process. In the first stage, the BEHAVE system was used to generate fire behavior predictions for a variety of fuel models under a range of wind and moisture conditions. These BEHAVE-generated observation were then used as inputs to BLDMDL as a test of the latter's ability to predict 'perfect' data. This procedure helps identify reasonable SIGMA, BETA, and MEXT search ranges and increment values. It also shows how well or poorly BLDMDL selects fuel parameters for fire behavior generated by fuel models with live fuels. tests were made using:

NFFL 01: Short grass (all dead fuel in one size class),

NFFL 11: Light slash (all dead fuel in three size classes), and

NFFL 04: Chaparral (dead and live fuel in three size classes).

In the second stage, BLDMDL was used to develop fuel models based upon actual fire behavior observations. Omi's (1986) observations in four grass-sage communities were initially chosen from among several available published data sets.

Several passes of BLDMDL were usually made to zoom in on a fuel model for a particular input set. The initial pass would usually identify several viable values for each parameter, which would be explored further in subsequent passes.

The processing logs for all input data sets and passes are appended.

The following observations were made during testing:

1 Optimal search increments are 100 (1/feet) for SIGMA, 0.0001 for BETA, and 0.01 for MEXT. Using these increments simultaneously may be prohibitive on a PC class machine.

2 Often many fuel parameter sets (using the increments specified in 1 above) are nearly equally suited for predicting the observed fire behavior. Differences in correlation coefficients or standard errors of the estimate between the number 1 and number 100 fuel parameter set are often not found until the third or fourth decimal place.

3 BLDMDL is able to correctly select the NFFL 1 parameter set if MEXT of 0.12 is 'hit' by the search algorithm.

4 BLDMDL less accurately selects NFFL 11 fuel parameter values.

5 BLDMDL was even less accurate in selecting NFFL 4 fuel parameters.

## RECOMMENDATIONS

Recommendations for further research, testing, and development fall into three categories; further testing using both generated and actual fire behavior inputs, development of better or alternative search and selection algorithms, and development of a better user interface including graphics and file import/export. Specifically:

### Algorithm Testing

1 Determine best ranges and increments for SIGMA, BETA, and MEXT search algorithm.

2 Perform time trials on selected computer equipment.

3 Test ability to select NFFL models with dead fuels only.

4 Test ability to select NFFL models with live fuels.

5 Test ability to select fuel models for actual fire observation data.

### Algorithm Extensions

1 Consider use of predicted vs. observed regressions forced through the zero intercept.

2 Add live fuels to the search parameters.

3 Explore alternative methods of identifying the 'best' fuel parameter set. For example, select the set with the most central values, but still within some limits of test statistic degradation.

4 Add capability of selecting two fuel models to predict an observation set.

5 Explore possibility of selecting models with several fuel strata to predict an observation set.

6 Add pre-search ability to preview fire behavior input observations for errant or contradictory behavior.

7 Add post-search analysis of input observations, recommending observations to omit or weight more heavily.

8 Add post-search distribution of fuel parameters into BEHAVE fuel categories and size classes.

9 Add ability to zoom in on a fuel parameter set.

## User Interface

1 Add interactive keyword or menu user interface, similar to BEHAVE.

2 Add graphics to display (1) predicted vs. observed fire behavior, and (2) behavior of the selected fuel model, with other NFFL or custom models, over a range of wind, slope, and fuel moisture conditions.

3 Add ability to post the selected fuel model to a BEHAVE fuel file, and conversely, to extract fuel models from the file into the graphics module.

## LITERATURE CITED

Omi, Philip N., and Mary M. Kwart. 1986.
Testing the BEHAVE Fire Prediction and Fuel Modeling System. Final Report on file at Intermountain Forest Experiment Station. 54 p.

Rothermel, Richard C. 1972.
A mathematical model for predicting fire spread in wildland fuels.